Online Subspace Estimation and Tracking from Incomplete and Corrupted Data

Laura Balzano
sunbeam@ece.wisc.edu

work with Robert Nowak, Benjamin Recht, Jun He (Nanjing UIST), and Arthur Szlam (CUNY)
Subspace Representations

Monitor/sense with $n$ nodes

$v \in \mathbb{R}^n$ is a snapshot of the system state (e.g., temperature at each node)

$v \in \mathbb{R}^n$ is a snapshot of the system state (e.g., traffic rates at each monitor)
\( v \in \mathbb{R}^n \) is a snapshot of the system state (e.g., temperature at each node)

\( v \in \mathbb{R}^n \) is a snapshot of the system state (e.g., traffic rates at each monitor)
Subspace Representations

Monitor/sense with $n$ nodes

Each snapshot lies near a low-dimensional subspace

$S \subset \mathbb{R}^n$

$v \in \mathbb{R}^n$ is a snapshot of the system state.

Using the **subspace as a model** for the data, we can leverage these dependencies for detection, estimation and prediction.
• For each frame we have \( n \) pixels.
• The background of a collection of frames lies in a low-dimensional subspace, possibly time-varying.
Suppose we receive a sequence of length-\(n\) vectors that lie in a \(d\)-dimensional subspace \(S\):

\[
v_1, v_2, \ldots, v_t, \ldots, \in S \subset \mathbb{R}^n
\]

And then we collect \(T\) of these vectors into a matrix,

\[
X = \begin{bmatrix}
v_1 & v_2 & \ldots & v_T
\end{bmatrix}
\]

If \(S\) is static, we can identify it as the column space of this matrix by performing the SVD:

\[
X = U\Sigma V^T.
\]

The orthogonal columns of \(U\) span the subspace \(S\).
Subspace Identification: Introduction

Suppose we receive a sequence of incomplete length-$n$ vectors that lie in a $d$-dimensional subspace $S$, and $\Omega_t \subset \{1, \ldots, n\}$ refers to the observed indices:

$$v_{\Omega_1}, v_{\Omega_2}, \ldots, v_{\Omega_t}, \ldots, \in S \subset \mathbb{R}^n$$

And then we collect $T$ of these vectors into a matrix:

$$X = \begin{bmatrix} v_{\Omega_1} & v_{\Omega_2} & \ldots & v_{\Omega_T} \end{bmatrix}$$

If $S$ is static, we can identify it as the column space of this matrix by performing the SVD:

$$X = U\Sigma V^T.$$  

The orthogonal columns of $U$ span the subspace $S$.  

Related Work: LMS subspace tracking

- Subspace tracking (with complete data) was approached with LMS methods in the 80s and 90s
  - Yang 1995, Projection Approximation Subspace Tracking; proof Delmas Cardoso 1998
  - Comon, Golub survey 1990

\[ \|v - P_S v\|_2^2 \]

- Incremental gradient methods are getting attention for their speed and convergence guarantees
  - Bertsekas, Tsitsiklis 2000
Residual with Incomplete Data

$U$ is an $n \times d$ orthogonal matrix whose columns span the $d$-dimensional subspace $S$. $U_\Omega$ denotes the submatrix with rows indicated by $\Omega$, where $\Omega \subset \{1, \ldots, n\}$ is the subset of indices observed.

**Full-data Residual**

$$P_S = U (U^T U)^{-1} U^T :$$

$$v_\perp = v - P_S v$$

**Incomplete-data Residual**

Let $P_{S\Omega} = U_\Omega \left(U_\Omega^T U_\Omega\right)^{-1} U_\Omega^T$.

$$v_\perp = v_\Omega - P_{S\Omega} v_\Omega$$
**Theorem: Incomplete Data Residual Norm**

$S$ is a known $d < n$ dimensional subspace of $\mathbb{R}^n$ with coherence $\mu(S)$.

$v_\Omega$ is our observation and we wish to estimate $\|v_\perp\|_2^2 = \|v_\Omega - P_S v_\Omega\|_2^2$

**Theorem:** If $|\Omega| = O(\mu(S)d \log d)$ and $\Omega$ is chosen uniformly with replacement, then with high probability and ignoring constant factors,

$$\frac{|\Omega| - d\mu(S)}{n} \|v - P_S v\|_2^2 \leq \|v_\Omega - P_S v_\Omega\|_2^2 \leq \frac{|\Omega|}{n} \|v - P_S v\|_2^2$$
Subspace Tracking

Suppose we receive a sequence of incomplete vectors that lie in a $d$-dimensional subspace $S$:

$$v_{\Omega_1}, v_{\Omega_2}, \ldots, v_{\Omega_t}, \ldots$$

Given $S_t$ and $v_{\Omega_t}$, how do we generate $S_{t+1}$?

Choose $S_{t+1}$ to decrease the error $\|v_{\Omega} - P_{S_{\Omega}} v_{\Omega}\|_2^2$. 
GROUSE

- Given step size $\eta_t$, subspace basis $U_t \in \mathbb{R}^{n \times d}$, observations $v_{\Omega_t}$
- Calculate Weights: 
  $$w = \arg \min_a \| U_{\Omega_t} a - v_{\Omega_t} \|_2^2$$
- Predict full vector: $v_\parallel = U_t w$
- Compute Residual on observed entries: $v_\perp = v_{\Omega_t} - (v_\parallel)_{\Omega_t}$ and zero-pad.
- Update subspace:
  $$U_{t+1} = U_t + \left( \sin(\sigma \eta_t) \frac{v_\perp}{\|v_\perp\|} + (\cos(\sigma \eta_t) - 1) \frac{v_\parallel}{\|v_\parallel\|} \right) \frac{w^T}{\|w\|}$$
  where $\sigma = \|v_\perp\| \|v_\parallel\|$.
- One iteration involves a projection and an outer product.
- The algorithm is simple and fast.
To use GROUSE, we pass over the columns of the matrix a few times in random order, doing an update for every column.

We compared against other state of the art MC algorithms on reconstruction error and computation time.
Robust Low-Rank Modeling (Robust PCA)

- Sparse + Low-Rank Model

Several algorithms have been developed to find such a decomposition from a matrix observation:
- convex optimization and approximations
GROUSE to GRASTA with Jun He and Arthur Szlam

\[ F_{grouse}(S; t) = \min_w \|U_{\Omega_t}w - v_{\Omega_t}\|_2^2 \]

\[ F_{grasta}(S; t) = \min_w \|U_{\Omega_t}w - v_{\Omega_t}\|_1 \]

\[ U_{t+1} = U_t + \left((\cos(\sigma\eta_t) - 1)U_t \frac{w_t}{\|w_t\|} + \sin(\sigma\eta_t) \frac{\Gamma}{\|\Gamma\|}\right) \frac{w_t^T}{\|w_t\|} \]
GRASTA Performance on RPCA with Jun He

- Simulated 2000 x 2000 matrix, rank=5
- Compare GRASTA to Inexact Augmented Lagrange Multiplier Method for RPCA

<table>
<thead>
<tr>
<th>% outliers</th>
<th>noise var</th>
<th>GRASTA 100% sampled</th>
<th>GRASTA 30% sampled</th>
<th>IALM FOR RPCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0.5e-3</td>
<td>3.64e-4 / 58.31 sec</td>
<td>6.07e-4 / 20.79 sec</td>
<td>16.7e-4 / 93.16 sec</td>
</tr>
<tr>
<td>10%</td>
<td>1.0e-3</td>
<td>7.643-4 / 59.55 sec</td>
<td>12.3e-4 / 20.66 sec</td>
<td>36.4e-4 / 117.76 sec</td>
</tr>
<tr>
<td>30%</td>
<td>0.5e-3</td>
<td>6.13e-4 / 67.19 sec</td>
<td>12.6e-4 / 22.63 sec</td>
<td>26.4e-4 / 324.26 sec</td>
</tr>
<tr>
<td>30%</td>
<td>1.0e-3</td>
<td>9.87e-4 / 69.06 sec</td>
<td>19.3e-4 / 22.85 sec</td>
<td>56.2e-4 / 362.62 sec</td>
</tr>
</tbody>
</table>
GRASTA Performance on Background Subtraction with Jun He

We used GRASTA for tracking experiments. When the background is dynamic, we use the full GRASTA Algorithm. Here we use three different resolution video datasets. Table 9 presents the approach for the various video datasets. They are the first, 7 frames. The subspace dimension is 5, for all. They are chosen randomly from throughout, and in the last three frames, they are the first two with static background and the last three with dynamic background video experiments. We used 0.7w of the pixels of every frame to do this update and we run the full GRASTA Algorithm. GRASTA tracks the subspace throughout the video (that is, the in the video is changing throughout. We use the “Lobby” dataset for tracking and separation only from 877w pixels to the right at each time (the middle row is the recovered background at each time after the camera pans takes 9.55 seconds for total 8850 sec frames as can be seen in Figure 6). When the background and foreground separation only from 877w pixels5, the estimated by ADMM separation shows how GRASTA performs cleanly with Figure 8.5. Demonstration of panning the “virtual camera” right to left through the video to simulate camera panning. The idea of the virtual camera is illustrated in Figure 7.5. Real-time video background and foreground separation. 

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Resolution</th>
<th>Total Frames</th>
<th>Training Time</th>
<th>Tracking and Separating Time</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport Hall</td>
<td>144 × 176</td>
<td>3584</td>
<td>11.3 sec</td>
<td>20.9 sec</td>
<td>171.5</td>
</tr>
<tr>
<td>Shopping Mall</td>
<td>320 × 256</td>
<td>1286</td>
<td>33.9 sec</td>
<td>27.5 sec</td>
<td>46.8</td>
</tr>
<tr>
<td>Lobby</td>
<td>144 × 176</td>
<td>1546</td>
<td>3.9 sec</td>
<td>71.3 sec</td>
<td>21.7</td>
</tr>
<tr>
<td>Hall with Virtual Pan (1)</td>
<td>144 × 88</td>
<td>3584</td>
<td>3.8 sec</td>
<td>191.3 sec</td>
<td>18.7</td>
</tr>
<tr>
<td>Hall with Virtual Pan (2)</td>
<td>144 × 88</td>
<td>3584</td>
<td>3.7 sec</td>
<td>144.8 sec</td>
<td>24.8</td>
</tr>
</tbody>
</table>
GRASTA Performance on Background Subtraction with Jun He
GRASTA demo

• Written by Arthur Szlam at CUNY in Open CV
Incremental Gradient on the Grassmannian for Background and Foreground Separation in Subsampled Video
Jun He, Laura Balzano, and Arthur Szlam. To appear at CVPR, June 2012.

Online Robust Subspace Tracking from Partial Information

Online Identification and Tracking of Subspaces from Highly Incomplete Information
Laura Balzano, Robert Nowak, and Benjamin Recht. Allerton, September 2010.

High-Dimensional Matched Subspace Detection when Data are Missing
Laura Balzano, Benjamin Recht, and Robert Nowak. ISIT, June 2010.

THANK YOU!
Questions?
Theorem: Incomplete Data Residual Norm

**Full Theorem:** If $|\Omega| \geq \frac{8}{3} \mu(S) d \log(2d/\delta)$ and $\Omega$ is chosen uniformly with replacement, then with probability $1 - 4\delta$,

$$
\frac{|\Omega|(1 - \alpha) - d\mu(S)(1+\beta)^2}{n} \|v - P_Sv\|_2^2 \leq \|v - P_{S\Omega}v\|_2^2 \leq (1 + \alpha) \frac{|\Omega|}{n} \|v - P_Sv\|_2^2
$$

where we write $v = x + y$, $x \in S$, $y \in S_\perp$,

$$
\alpha = \sqrt{\frac{2\mu(y)^2}{|\Omega|} \log \left( \frac{1}{\delta} \right)}, \quad \beta = \sqrt{2\mu(y) \log \left( \frac{1}{\delta} \right)}, \quad \text{and} \quad \gamma = \sqrt{\frac{8d\mu(S)}{3|\Omega|} \log \left( \frac{2d}{\delta} \right)}.
$$

**Lemma 1:** $\|y_\Omega\|_2^2 \geq (1 - \alpha) \frac{|\Omega|}{n} \|y\|_2^2$ \quad **McDiarmid’s inequality with sum of RVs**

**Lemma 2:** $\|U_\Omega^T y_\Omega\|_2^2 \leq (1 + \beta)^2 \frac{|\Omega|}{n} \frac{d\mu(S)}{n} \|y\|_2^2$ \quad **McDiarmid’s inequality**

**Lemma 3:** $\| (U_\Omega^T U_\Omega)^{-1} \|_2 \leq \frac{n}{(1 - \gamma)|\Omega|}$ \quad **Non-commutative Bernstein inequality**
Descent on the Grassmannian

• Idea: Stochastic gradient descent to minimize the incomplete project residual one vector at a time. (Subspace Tracking in Signal Processing is done this way, using the complete-data residual.)

• Since we are estimating a subspace, we can perform gradient descent directly on the Grassmanian manifold $G(n,d)$ and follow its geodesics.
  
  (There are explicit formulas for a gradient descent step that follows the Grassmannian geodesic.)


1-d subspaces in $\mathbb{R}^2$: 